Refine Search

Search Results -

Term	Documents
"5724600"	2
5724600S	0
SERVER	394652
SERVERS	109921
(("5724600".PN.) AND SERVER).PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	0
(5724600.PN. AND SERVER).PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	0

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L13

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Wednesday, April 26, 2006 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L13	5724600.pn. and server	0	L13
L12	6317807.pn. and server	0	L12
L11	L10 and (resource$ near allocation)	7	L11
L10	L9 and updat$	52	L10
L9	L8 and l1	110	L9
L8	disk near storage	59773	L8
L7	L6 and subsystem	12	L7
L6	L5 and replicat$	38	L6

L5	L4 and disk$	215	L5
L4	L3 and settings	559	L4
L3	L1 and specification	1516	L3
L2	L1 and spcification	0	L2
L1	transfer near capacity	6250	L1

END OF SEARCH HISTORY

L11: Entry 7 of 7 File: USPT Mar 3, 1998

DOCUMENT-IDENTIFIER: US 5724600 A
** See image for Certificate of Correction **
TITLE: Parallel processor system

Brief Summary Text (6):
Generally, in parallel processor system including a considerable large number of processors, it has been difficult that all the processors share directly an input and output unit (hereinafter, referred to as an I/O unit) such as a magnetic disk storage unit. It is required that data is exchanged between processors via another communication mechanism other than the I/O unit, e.g. a bus construction or communication network mechanism, without sharing the I/O unit among the processors. The present invention relates to a technology assuming an aspect in which a single I/O unit is not dynamically shared among plural processors. (however, the system and processor of the present invention (to be described later) can be used in the mode that a single I/O unit is shared among plural processors).

Brief Summary Text (39):
The skew of MIMD of the S-DPr is in substance an unavoidable skew, like the skew of the selection rate. Now, any systematic countermeasure has not been proposed. However, this problem is not important since performing preferentially the resource allocation as the S-DPr, compared with the resource allocation to the other operation, and reducing in concrete the skew of the MIMD of the S-DPr can be realized within the existing technology.

Detailed Description Text (59):
Particularly, the problem is the amount of the logical circuits except memories acting as necessary work memories. As to the memory capacity, in the existing technology, the total capacity of register regions within the switching unit is of the order of n.sup.2 log n where n is the number of processors. On the other hand, the total capacity of the transfer destination processor decision table 420 being an alternative of the present embodiment is of the order of n.sup.3. The difference in memory capacity expands rather. However, the cost to the capacity of the current memory is sufficiently low. On the other hand, the cost of the logical circuit, if the developmental cost is also taken into account, becomes incomparatively high to that of the memory, in use of the number of transistors with the same amount.

Detailed Description Text (61):
On the other hand, even if n is in the order of several hundreds, the capacity of the transfer destination processor decision table 420 is of the order of at most several MB in one processor. This does not correspond to a sole high-performance processor chip in cost. That is, according to the present embodiment, the system cost added does not reach the original cost on the processor side. Hence it is clear that there is an advantage in cost.

Detailed Description Text (85):
The transfer destination processor decision table 820 corresponds structurally to the table 420 in the first embodiment. The conversion one-dimensional list (not shown in the second embodiment) that converts the identifier of a column in the table 820 into the processor identifier of the T-DPr corresponds structurally to the list 421 in the first embodiment. However, there is a difference (to be described later) between the first and the second embodiments in the algorithm for updating the content in the table 820.

Detailed Description Text (105):
(4) The microprocessor 801 updates by adding the value of a specific factor in the table 820 specified by the abovementioned process and "1+the value of the queue length field 841 of a transfer queue (here, with numeral 830) corresponding to the destination T-DPr decided by the above-mentioned process" (the reason why one is added is that one tuple to be subjected directly to the destination decision process is enqueued in the post process) (step 801F; "Update Table 820").

Detailed Description Text (106):
As described above, in the updating process, the value of each element in the transfer destination processor decision table 820 is weighted proportionally to the operation performance of the corresponding one of the T-DPrs 900A to 900C and the weighted value becomes on index showing roughly that how large load regarding what bucket ID is added to each of the T-DPrs 900A to 900C. In the present embodiment, the value of each element in the transfer destination processor decision table 820 is controlled so as to be equalized as much as possible to every T-DPr in a bucket ID.

Detailed Description Text (119):
(3) Thereafter, the microprocessor 901 updates the value of the pointer 922 by incrementing by one (step 901B; "Update Round Robin Pointer"). As described above, when the value of the pointer 922 equals to (n-1), where n is the number of S-DPrs (the value of the machine number information field 923A) in the table 923, the value of the pointer 922 is reset to "0" by the microprocessor 901.

Detailed Description Text (135):
(3) Since the input/output register must have such a sufficient capacity that all the transfer units can be stored, blocking and transferring as described above means that hardware amount increases in all the switching units (refer to numerals 201 to 212 in FIGS. 37 and 38).

Detailed Description Text (138):
An increase in amount, of course, occurs in the average capacity of the transfer queues 830 to 832, the buffer capacity of the input port circuit 803 and output port circuit 804 in the processors 800A to 800C, and the buffer capacity of the input port circuit 903 and output port circuit 904 in the processors 900A to 900C. However, the increase in amount is less than at least 1/log n, compared with that in the existing technology. The adverse effect due to the fluctuations in the load to the network mechanism 600 is small. Hence it is possible to reduce the transfer process overhead.

Detailed Description Text (146):
When the transfer data (tuple group) prepared in the transfer buffer 850 is transferred, the microprocessor 801 also transfers the top address and the capacity of the transfer buffer 850 to the output port circuit 804.

Detailed Description Text (158):
When the task which prepares transfer data in the transfer buffer 850 has been completed, the microprocessor 801 reports and transfers the top address and the capacity of the transfer buffer 850 to the output port circuit 804.

Detailed Description Text (179):
(1) The data transfer is performed by using as one unit "a capacity within a fixed range" of which the upper limit and the lower limit are determined. (2) The load evaluation index or a value accumulated in the transfer destination processor decision table is changed from "the number of tuples in a transfer queue+1" in the third embodiment into "the tuple capacity in a transfer queue+the capacity of a tuple being currently processed" or "the tuple capacity in a transfer queue+the capacity of the transfer data corresponding to a tuple processed".

Detailed Description Text (186):
In the S-DPr 1200 in the fourth embodiment, the capture timing of the transfer buffer 1250 is the same as the timing of the transfer buffer 850 in the third embodiment. However, the microprocessor 1201 dequeues tuples of number satisfying the upper limit capacity value And the

lower limit capacity value regarding the transfer buffer 1250 (or the transfer data corresponding to the tuples), stores them into the transfer buffer 1250, and then prepares the transfer data. In this case, the transfer data format is the same as that in the third embodiment. In the fourth embodiment, the capacity (the capacity of the transfer buffer 1250 captured in the main storage unit 1202) is surely between the lower limit capacity value And the upper limit capacity value.

Detailed Description Text (199):
If the value of the transfer data amount field 1247 is more than the lower limit value of the transfer data amount, the microprocessor 1201 captures the transfer buffer 1250 into the main storage unit 1250. At this time, the capacity of the transfer buffer 1250 is the upper limit value of the transfer data amount.

Previous Doc Next Doc Go to Doc#